In the Claims

For the convenience of the Examiner, Applicant has produced all pending claims. The claims are not amended.

Claims 1-24 were previously cancelled without prejudice or disclaimer.

25. **(Original)** A method for forming an object-based computer system comprising:

providing a first existing executable module and a second existing executable module;

determining a first operation associated with the first existing executable module;

determining a second operation associated with the second existing executable module;

determining a mapping between the first and second operations; and

managing an interaction between the first and second operations based on the mapping.

26. **(Original)** The method according to Claim 25, wherein the first and second existing executable modules respectively comprise an executable component object.

27. **(Original)** The method according to Claim 26 and further comprising assembling the executable component objects to form an object-based application.

28. **(Original)** The method according to Claim 26 and further comprising managing runtime interactions between the executable component objects.

29. **(Original)** The method according to Claim 28, wherein managing the runtime interactions comprises configuring a user interface based on the mapping for managing the runtime interactions.

30. **(Original)** The method according to Claim 25, wherein determining the mapping comprises specifying an intermediate representation of information for communication between the first and second operations.

31. **(Original)** The method according to Claim 30, wherein the intermediate representation is associated with a user interface.

32. **(Original)** The method according to Claim 30, wherein the intermediate representation indicates how the first operation responds to a user interface event.

33. **(Original)** The method according to Claim 25, wherein determining the mapping comprises determining how a parameter associated with the first operation flows to the second operation.

34. **(Original)** The method according to Claim 25 and further comprising managing a data value associated with the first operation when the first operation is invoked.

35. **(Original)** The method according to Claim 25 and further comprising:
mapping an output parameter associated with the first operation; and
mapping an input parameter associated with the second operation.

36. **(Original)** The method according to Claim 25, wherein the first operation has an associated field and further comprising generating a characteristic associated with the first operation based on the field and user input.

37. **(Original)** The method according to Claim 25, wherein determining the mapping comprises determining a declarative mapping between a first parameter associated with the first operation and a second parameter associated with the second operation.

38. **(Original)** A system for forming an object-based computer system comprising:

a first existing executable module;

a second existing executable module;

means for determining a first operation associated with the first existing executable module;

means for determining a second operation associated with the second existing executable module;

means for determining a mapping between the first and second operations; and

means for managing an interaction between the first and second operations based on the mapping.

39. **(Original)** The system according to Claim 38, wherein the first and second existing executable modules respectively comprise an executable component object.

40. **(Original)** The system according to Claim 39 and further comprising means for assembling the executable component objects to form an object-based application.

41. **(Original)** The system according to Claim 39 and further comprising means for managing runtime interactions between the executable component objects.

42. **(Original)** The system according to Claim 41, wherein the means for managing the runtime interactions comprises means for configuring a user interface based on the mapping for managing the runtime interactions.

43. **(Original)** The system according to Claim 38, wherein the means for determining the mapping comprises means for specifying an intermediate representation of information for communication between the first and second operations.

44. **(Original)** The system according to Claim 38, wherein means for determining the mapping comprises means for determining how a parameter associated with the first operation flows to the second operation.

45. **(Original)** The system according to Claim 38 and further comprising:
means for mapping an output parameter associated with the first operation; and
means for mapping an input parameter associated with the second operation.

46. **(Original)** The system according to Claim 38, wherein the first operation has an associated field and further comprising means for generating a characteristic associated with the first operation based on the field and user input.

47. **(Original)** The system according to Claim 38, wherein the means for determining the mapping comprises means for determining a declarative mapping between a first parameter associated with the first operation and a second parameter associated with the second operation.

48. **(Original)** A system for forming an object-based computer system comprising software stored on storage and operable to:

provide a first existing executable module and a second existing executable module;

determine a first operation associated with the first existing executable module;

determine a second operation associated with the second existing executable module;

determine a mapping between the first and second operations; and

manage an interaction between the first and second operations based on the mapping.

49. **(Original)** A method for forming an object-based computer system comprising:

providing a first existing executable component object and a second existing executable component object;

determining a first operation associated with the first existing executable component object;

determining a second operation associated with the second existing executable component object;

mapping an output parameter associated with the first operation to an input parameter associated with the second operation;

managing the flow of the output parameter to the input parameter based on the mapping;

assembling the first and second executable component objects to form an object-based application;

configuring a user interface based on the mapping for managing the runtime interactions between the output parameter and the input parameter; and

managing a data value associated with the first operation when the first operation is invoked.